

# Segmenting Time Series for Weather Forecasting

Somayajulu G. Sripada, Ehud Reiter, Jim Hunter, and Jin Yu  
Dept of Computing Science  
University of Aberdeen  
Aberdeen, UK  
{ssripada,ereiter,jhunter,jyu}@csd.abdn.ac.uk

Abstract

We are investigating techniques for producing textual summaries of time series data. Deep reasoning techniques have proven impractical because we lack perfect knowledge about users and their tasks. Data analysis techniques such as segmentation are more attractive, but they have been developed for data mining, not for communication. We examine how segmentation should be modified to make it suitable for generating textual summaries. Our algorithm has been implemented in a weather forecast generation system.

## 1. Introduction

Textual summaries of time series data, either independently or in conjunction with graphical presentation, aid human understanding of the underlying data sets. SUMTIME is a research project aiming to develop a generic model for summarisation of time series data. Initially, we have applied standard knowledge acquisition (KA) techniques such as think aloud sessions and corpus studies [1] to study how humans carry out the task. These KA studies helped us in specifying the requirements for data summarisation.

We found three alternative ways in which we could model data summarisation. One approach is based on deep reasoning methods suggested by [2]. This model explicitly reasons with user tasks and knowledge to determine information to include in the text summary. But in practice this is not attractive for two reasons. The first reason is that deep reasoning methods are computationally intensive. The second is that deep reasoning methods require precise knowledge of the user's tasks and goals. However, our KA studies have shown that it is not possible to get such knowledge. For example, meteorologists might know that the user wants to carry out supply boat operations alongside the rig structure, but would only know broadly the operational procedures followed by the oil company staff to carry out this task, or the precise operational characteristics of specific supply boats.

Alternatively, a model of data summarisation using existing models of data analysis is feasible. However, as explained in the paper, such a model does not agree with the observations made in our KA studies of humans summarising data. Therefore, in SUMTIME we have tried to adapt existing data analysis models so that the

adapted model agrees with our KA observations. The result is a model, which has some sensitivity to end-user needs but does not reason from first principles.

There are two perspectives to the current work. From the perspective of communicating summaries to end users, this work offers the alternative approach of adapting existing data analysis techniques to the approach of reasoning from first principles. From the perspective of data analysis, the work addresses the issues involved in adapting the existing techniques for use in a communicative context.

## 2. Background

SUMTIME is an ongoing research project aiming to develop generic techniques for producing textual summaries of time series data. We focus our study on time series data derived from the domains of meteorology and gas turbines. In the domain of meteorology, time series data produced by numerical weather prediction (NWP) models is summarised as weather forecast texts. In the domain of gas turbines, sensor data from an operational gas turbine is summarised for the maintenance engineers. More details on SUMTIME have been described in [3].

day	hour	wind dir	wind speed (Knots)
20-1-01	6	S	4
20-1-01	9	S	6
20-1-01	12	S	7
20-1-01	15	S	10
20-1-01	18	S	12
20-1-01	21	S	16
21-1-01	0	S	18

Table 1: Wind predictions from numerical model on 20-01-01

Table 1. shows an example time series from the domain of meteorology. It shows the wind speed and wind direction data predicted by the numerical model for the forecast period 0600-2400 hours on 20 Jan 2001. It is a time series sampled every three hours.

---

06-24 GMT, 20- Jan 2001:  
S 02-06 INCREASING 16-20 BY EVENING

---

Figure 1. Wind texts from human written forecast for 20-01-01

Figure 1. shows the human written summary for the data shown in Table 1. The wind text is part of a marine forecast issued by our collaborating organisation, WNI/Oceanroutes, Aberdeen for offshore oil company staff. There are a number of tasks performed by the oil company staff, such as flaring the excess gas and carrying out supply boat operations that depend upon the condition of the weather. Marine forecasts are required to keep the oil company staff informed about the

weather conditions so that they can make right decisions about their activities. Human forecasters use their experience to produce weather reports that are useful to the oil company staff.

The first phase of SUMTIME focused on studying how humans carry out summarisation. During this phase, we applied well-known knowledge acquisition (KA) techniques [4] and recorded a number of observations about human data summarisation [1]. The second phase is responsible for formalising the observations and for building a model for data summarisation as guided by the observations. In terms of the phases of a software life cycle, the first phase of SUMTIME corresponds to the requirements gathering and the second phase to the design of software to suit the requirements.

One of the objectives of SUMTIME has been to bring together two areas of research, time series data analysis and text generation. Accordingly, our model building activity involved selection of existing models from the above two fields that satisfy our observations. Before we discuss this any further, we need to introduce a few ideas from the field of text generation.

## 2.1 Review of Text Generation

Text generation is the study of developing computational models for producing natural language (say English) texts from computer internal representations (say records in a database). Figure 2 shows three-stage pipeline reference architecture for text generation systems [5].

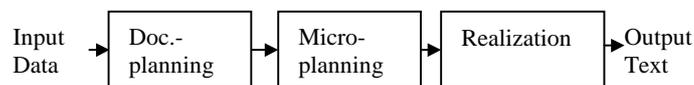


Figure 2. Three-stage pipeline reference architecture of a Text Generator

Document-Planning – Also known as content planning, this stage is responsible for selecting the content that needs to be expressed in language and for organising them coherently.

Micro-Planning – this stage decides how to express information linguistically, for example what words should be used.

Realisation – this stage produces grammatical output text according to the syntax and morphology of the target language.

The focus of this paper is on the document-planning stage, particularly on the task of selecting content for summarisation. Empirical observations from KA have guided us in our search for a suitable model.

## 2.2 Content Selection and Time Series Data Analysis

In our case, where the main task is to summarise the input data set, content selection involves rejecting ‘unimportant’ portions of the input. Of course, deciding

what is important depends upon the end user. In practice, applying notions of importance directly to the raw data might become computationally inefficient as the size of the input data set increases. A more attractive option is to process the raw data using data analysis techniques and to apply the importance constraints to the pre-processed data. Evidently, there are two subtasks involved here. The first is concerned with processing the raw data and the second task with application of the importance criteria. Interestingly, our observations from KA have supported this approach.

One of the experts, with whom we did think aloud sessions, suggested what we call ‘step model’ to decide what data items to include in the summary. More precisely, we define a step value for each data channel, which may depend on the end user. We start processing by selecting the first point as the anchor. We then compute for each successive point in the input its difference with the anchor until the difference value for a point exceeds the step value. This point (where the difference exceeded the step value) is then set as the new anchor and the above process of computing differences is repeated for the remaining points. Finally, all the anchor values are selected for inclusion in the summary. However, our corpus studies in the domain of meteorology showed that human forecasters do not follow the simple step model.

For example, the wind data shown in Table 1, shows wind speed values monotonically increasing from 4 to 18. The step model with a step value of 5 Knots selects wind speed values of 4, 10 and 16. The generated summary is shown in Figure 3. We follow human forecasters and do not mention wind speed values literally in the summary. Instead, we construct a range around the speed (e.g. 3-8 instead of 4) and mention it in the summary.

---

06-24 GMT, 20- Jan 2001:  
S 3-8 INCREASING 8-13 BY AFTERNOON AND 13-18 BY EVENING.

---

Figure 3. Wind texts from Step Model for the data shown in Table 1.

This of course differs from the actual forecast text shown in Figure 1. In this case and many other similar cases we have noticed that human forecasters seem to carry out a task known as ‘segmentation’ in time series data analysis. Segmentation refers to the process of approximating a time series of length  $n$  with  $K$  straight lines, where  $K$  is much smaller than  $n$ . The approximated signal is also known as a piecewise linear representation of the input time series. Segmentation helps to split the input data set into a number of intervals.

There have been a few attempts at using data analysis techniques for summarising time series data earlier [6][7]. TREND [6] used wavelet theory to analyse weather data (not predicted data as we use, but archived weather measurements) and produced summaries describing the weather conditions that existed in the time period covered by the data. ANA [7] uses a combination of arithmetic

computations and pattern matching techniques to analyse raw data from the Dow Jones News service database.

Initially, in SUMTIME, we implemented the segmentation algorithms as variously reported [8][9][10]; we also implemented the simple step model suggested by the above-mentioned expert. When we tested the algorithms on real data, we observed a number of ways in which the standard algorithms did not match what the human forecasters did. But before we describe the impact of our observations on our model building activity we need to give a brief account of segmentation algorithms.

### 2.3 Review of segmentation algorithms

The exact descriptions of segmentation algorithms are not given here as they are available in full detail elsewhere [9][10]. The main objective of this section is to introduce the important concepts related to popular segmentation algorithms.

There are three types of segmentation algorithms [9], sliding window, top-down and bottom-up. This is based on whether the segments are formed left to right (iteratively extending the segment in the current window to include more data points), top-down (breaking down the series iteratively into ever-smaller segments) or bottom-up (merging neighbouring segments iteratively into ever-larger ones). Each offers different advantages over the others and therefore choosing the right algorithm is an important task.

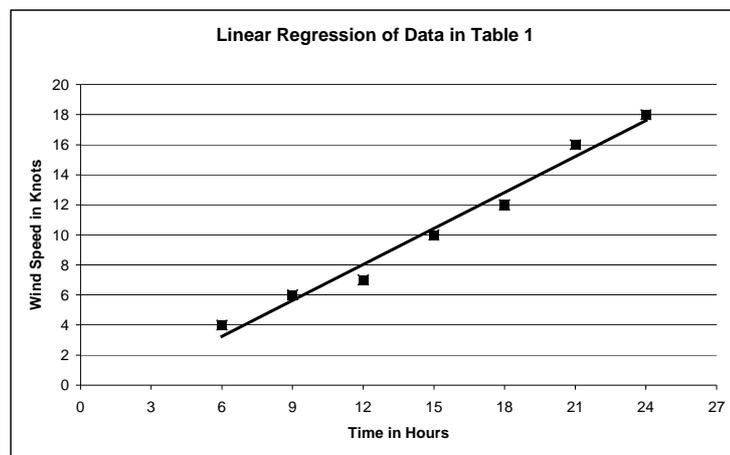
There is another choice to be made when using segmentation algorithms. This is with regard to the choice of which approximating line (or curve) we use to fit the data. We could use linear regression or linear interpolation (line joining the end points of a sub-series). Linear regression as shown in Figure 4a tries to fit a line for the data points using the least square error measure. Linear interpolation as shown in Figure 4b simply joins the end points in a data set with a straight line. Though the algorithms mentioned above can be used with either of these approximating lines, this choice is important from the perspective of communication as explained in the next section.

Finally, while using segmentation algorithms another important issue is related to the results of segmentation, i.e. the number of segments produced. Segmentation algorithms need a stopping criterion to terminate iteration. In its most common form [9], segmentation problem is framed as:

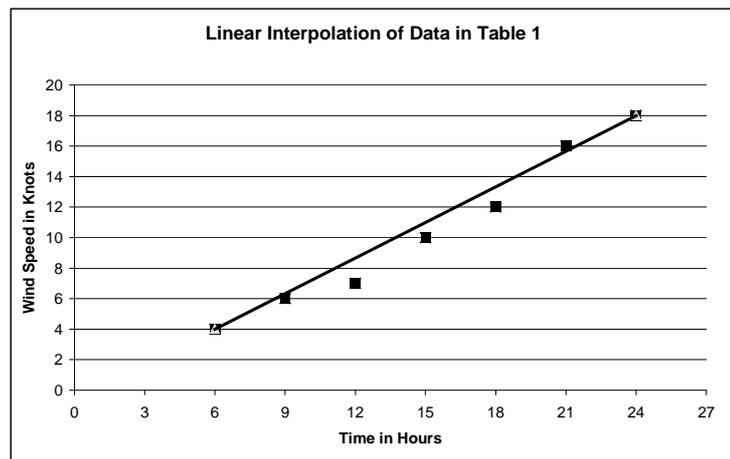
Given a time series  $T$ , produce the best representation either

- using exactly  $K$  segments or
- such that the maximum error for any segment does not exceed some user-specified threshold,  $\text{max\_error}$  or
- such that the combined error of all segments is less than some user-specified threshold,  $\text{total\_max\_error}$ .

The above problem specification presents three ways in which we can specify stopping criterion. The first always produces a predefined number of segments. The other two produce segments that fit the data without exceeding the specified error thresholds. For the purpose of this paper, it is important to distinguish two types of stopping criteria: internal (data dependent) and external (user dependent). For example, [8] shows that Balance of error when used as an internal stopping criterion can produce an ‘optimum’ number of segments for the given data set. For communication, as discussed in section 3.4, external stopping criteria are important.



(a) Linear Regression



(b) Linear Interpolation

Figure 4. Approximating Lines for Wind Speed Data in Table 1

### **3. Requirements due to Communication**

Segmenting time series data is carried out for achieving many objectives. For example, segmented time series can be used as a concise representation in data mining applications [8][9]. In this section we describe a few issues we observed from our KA that have impact on either the choice of the segmentation algorithm for communication or on the way it is implemented. These observations were based on a variety of KA techniques, including

- analysis of a corpus of human-written forecasts
- a think-aloud session with an expert forecaster
- observing forecasters as they worked
- asking forecasters what they did
- discussions with forecasters about our observations.

#### **3.1 Importance is Relative**

Humans like to say something, which summarises the input data even when there is nothing really important detected in the input data. In the domain of meteorology, the forecast for a quiet day carries minor weather variations, which would not be mentioned in a forecast for a 'bad weather' day. A similar observation was made in [11].

#### **3.2 Importance of Error Varies**

Different ranges of data have different impact on the end user. For example, weather forecasts aid oil company staff in deciding when to perform tasks. At low wind speeds, the wind direction has little impact on operations. So small changes in wind direction do not need to be reported. But at high wind speeds, small changes in wind direction assume significance and therefore need to be reported. This implies that the importance of error introduced by segmentation varies with the range in which the values fall. For the above case, we can tolerate larger error in segments that approximate wind direction series at low wind speeds, but not at high speeds. Also, the definitions of these ranges vary with the end user.

#### **3.3 Channels need Co-ordination**

Because of the interrelationships among the various data channels, processing these channels needs co-ordination. For example, wind text is produced by processing data sets of wind speed and wind direction. The results of the processing from the wind speed need to be co-ordinated with that from the wind direction.

For example, the text shown in Figure 5 suggests that the wind veers (moves clockwise) N and eases to 14-18 at late evening, i.e. 2400 hours. But as can be seen

from Table 2, although the speed change does indeed happen at 2400 hours, the direction change in fact takes place at 1800 hours.

day	hour	wind dir	wind speed (Knots)
16-9-01	6	NNW	24
16-9-01	9	NNW	24
16-9-01	12	NNW	24
16-9-01	15	NNW	24
16-9-01	18	N	24
16-9-01	21	N	20
17-9-01	0	N	16

Table 2: Wind predictions from numerical model on 16-09-01

---

06-24 GMT, 16- Sept 2001:  
NNW 22-26 VEERING N EASING 14-18 BY LATE EVENING

---

Figure 5. Wind texts from human written forecast for 16-09-01

### 3.4 Stopping Criterion Sensitive to End User

We believe that summarisation of time series data requires that we pick up those elements of the input data set that are representative of the data and also in some sense ‘important’ to the end user of the summaries. Therefore, the stopping criterion for the segmentation process cannot depend upon the properties of the data set alone. Instead the segmentation should be sensitive to the end user too. For example, the preferred wind details for an oilrig in the North Sea differ from those for an oilrig in the Persian Gulf due to the differences in the rig design or construction.

### 3.5 Linear Interpolation as Approximating Line

The data values mentioned in the summary have to be present in the input data set because this is how humans (as we have observed with expert forecasters) write summaries. Therefore, linear interpolation seems to be a better option than linear regression as the approximating line.

### 3.6 Impact of Overview

One of the observations from our KA exercises revealed that humans build an overview of the input data set before writing summaries. Based on this observation, we have proposed a two-stage model for content determination, which has been described in [12]. In view of this, sliding window algorithm is not suitable because the algorithm lacks a global view (overview) of the data [9].

## 4. SUMTIME-MOUSAM

The implementation phase of SUMTIME has been devoted to building a test bed system we call SUMTIME-MOUSAM [3]. The main design objective of this system has been to assemble all the support components needed to experiment with summarisation of time series data so that adding newer components to the system involves reduced effort. It consists of a database of time series data, data processing utilities, a corpus of human written summaries, tools to parse human written summaries into conceptual representations, comparison utilities, and a micro-planner and sentence realiser (see section 2.1). This section focuses on the implementation of segmentation in SUMTIME-MOUSAM based on the observations presented in Section 3. As stated earlier, segmentation is used here to pre-process data for summarisation.

As discussed in section 3.6, segmentation results would be more suited for communication if the algorithm has a global view of the entire data set. The bottom-up algorithm has been chosen in SUMTIME-MOUSAM because of its ability to look globally while segmenting. Also, as discussed in section 3.5, linear interpolation is more suited for our purpose and has been chosen as the approximating line. Having made these two decisions, we still needed to decide about the stopping criterion for the bottom-up algorithm as described in [9][10].

As described earlier in section 2.3, the stopping criterion dictates when the segmentation process should stop. This depends upon a number of issues. Section 3.1 argues in favour of a minimum number of segments if the results are to be used for summarisation. Section 3.2 suggests that the segmentation should proceed to different levels of granularity based on the ranges of values underlying the segments and therefore indirectly suggests that the stopping criterion needs to be different for different ranges of values in the input data. Finally as argued in section 3.4, the stopping criterion should be derived from end user preferences.

We have created a table (acquired from our KA) of end user preferences that contains the stopping criterion as a threshold value. Table 3 shows an example table compiled for one particular end user. This table is used for segmenting wind speed and wind direction data. Here, the first column shows the different speed ranges. For each of these ranges, the second column shows the stopping thresholds for segmenting direction data. The third column shows the thresholds for segmenting speed data. Though in this table the speed thresholds are same for all the ranges, they need not be so. It can also be noted from the table that segmenting direction data depends upon ranges of wind speeds. This partially addresses the interdependence among various channels, which has been described in section 3.3. We have not yet completely addressed this issue in our work.

Wind Speed	Direction Threshold	Speed Threshold
0 - 15	44	5
15 - 40	22	5
40 - 65	22	5
> 65	22	5

Table 3: Example table showing the ranges and thresholds for a particular oilrig.

In SUMTIME-MOUSAM, we have used the bottom-up segmentation algorithm as described in [9]. We have used linear interpolation as the approximating line and computed the measure of goodness of fit as the maximum of the set of vertical distances from the numerical data values to the approximating line. Table 3 is then used to read range sensitive threshold values as stopping criteria. For example, in the speed range of 0-15 Knots, segmentation of wind direction data can proceed until the maximum distance exceeds 44 degrees. Figure 5 shows the summary produced by the segmentation model. It can be noted that instead of mentioning two stage increase in the wind speed as done by the step model (see the wind text in Figure 3), the segmentation model mentions the whole increase as one change, which is very similar to the human summary (see the wind text in Figure 1).

SUMTIME-MOUSAM also performs microplanning and realisation. For example, in Figure 6 the microplanner has decided to realise low speeds without a leading zero (so, "3-8" instead of "03-08"), in contrast to what the human forecaster did in Figure 1, because leading zeros were usually omitted in our corpus. Perhaps more interestingly, the microplanner has also decided to ellide (eliminate) the end-of-segment time phrase ("by evening" in Figure 1); this is again based on our corpus study, which suggests that forecasters usually (but not always) omit the end-of-segment time phrase when a text describes a single segment. SUMTIME-MOUSAM's realiser has chosen the order of the words in the text (which is straightforward in this example); for example the realiser knows that direction usually precedes speed ("S 3-8" instead of "3-8 S").

---

06-24 GMT, 20- Jan 2001:  
S 3-8 INCREASING 15-20.

---

Figure 6. Wind texts from Segmentation Model for the data shown in Table 1.

## 5. Evaluation

In this section we describe evaluation of SUMTIME-MOUSAM. Ideally we would like to evaluate our system based on how useful the end users (oil company staff) find our summaries for carrying out their tasks (such as flaring excess gas and carrying out supply boat operations). Such task-based evaluation can be difficult to organise and expensive to perform [13]. We intend to carry out a task-based evaluation in the future, but initially we instead performed an intrinsic evaluation as described below. This compared textual summaries produced by using the step method (described in section 2.2) and the segmentation method (described in section 4).

The primary objective of data summarisation is to increase what we call 'accessibility' of the input data set by extracting one of its subsets which is shorter than the original data set and also contains important information from the input data set. In other words, size and 'informativeness' are two important features of a summary; the smaller the size and the more informative the content, the better the summary. Our evaluation uses the error introduced by the data analysis methods (a

measure of information loss) as one of the metrics for comparison. The other metric we use is related to the size of the summary. Because we were evaluating content algorithms and not realisation or microplanning, we used a semantic measure of size instead of counting words or characters. This measure was the number of "tuples" in the summary, which essentially corresponds to the number of wind states mentioned in the summary.

Consider for example the wind summary text from the segmentation model shown in Figure 6. The text mentions two wind states "S 3-8" and "15-20", and hence its tuple count is 2. The example summary text from the step model shown in Figure 3 mentions three wind states ("S 3-8", "8-13", and "13-18"); hence its tuple count is 3.

In the case of error metric, the computation of errors is complicated by the fact that there are two parameters, wind speed and wind direction. We computed the individual errors for speed and direction, normalised these using error thresholds similar to the ones (Speed threshold and direction threshold) shown in Table 3, and then added these to get a total error.

For our evaluation study we have produced 357 forecasts using both the step method and the segmentation method. We computed the following metrics for each of the forecasts:

Et – Error due to step method

Es – Error due to segmentation method

Tt – number of tuples from the step method

Ts – number of tuples from the segmentation method.

These metrics are then used to compare the two methods. Initially we compared the two methods using the individual metrics. For this we defined a number of simple propositions such as 'Et>Es' (Is error due to step method greater than the error due to segmentation method?). For each such proposition, we have counted how many times it is true. The results of evaluation for the individual metric comparisons have been shown in Table 4.

Et>Es	Et<Es	Et=Es	Tt>Ts	Tt<Ts	Tt=Ts
(1)	(2)	(3)	(4)	(5)	(6)
180	172	5	175	11	171

Table 4: Results of Evaluation – individual metric comparisons

This table suggests that the step and segmentation models lead to similar error rates but that the segmentation model produces shorter texts. In other words, segmentation is producing shorter texts than the step model without losing accuracy. This is indeed what we see in the example texts shown in Figures 3 (step model) and 6 (segmentation model), which are based on the same data: the segmentation model text is shorter than the step text, but almost as accurate.

Et>Es & Tt>Ts (1)	Et<Es & Tt<Ts (2)	Et>Es & Tt<Ts (3)	Et<Es & Tt>Ts (4)	Et<Es & Tt=Ts (5)	Et>Es & Tt=Ts (6)	Et=Es & Tt<Ts (7)	Et=Es & Tt=Ts (8)	Et=Es & Tt>Ts (9)
59	2	9	114	56	112	0	3	2

Table 5: Results of Evaluation – combined metric comparisons

Table 5 shows a more detailed analysis of our experimental results. We see that the segmentation method is better than the step method in both size and error (Proposition (1), (Et>Es & Tt>Ts)) in 16.5% of cases. On the other hand, the step method is better than the segmentation method in both size and error (Proposition (2), (Et<Es & Tt<Ts)) in only 0.56% of cases. We can draw the following conclusions as well:

2.5% of cases segmentation is better error wise but worse size wise (proposition 3)

32% of cases segmentation is better size wise but worse error wise (proposition 4)

31% of cases segmentation is better error wise but equal size wise (proposition 6)

An initial run of our evaluation in fact highlighted one problem with segmentation, which occurs when the wind is fairly steady over the course of a forecast period. In such cases the original segmentation model produced one segment and hence two tuples, while the step model produced only one tuple (since the wind didn't change enough to cross a threshold). For example, if the wind rose from 10 to 12 over the course of a period, and was from the N during the entire period, the original threshold model produced "N 8-13 RISING 10-15", whereas the step model produced just "N 8-13". We examined our corpus of human texts and discovered that in such cases human authors described the wind with one tuple, that is they did not mention very small 2 knot rises in the wind speed. We accordingly changed our segmentation model so that in such cases it also produced one tuple. This observation opened up the possibility of finding more modifications to our algorithm.

## 6. Segmenting Sensor Data from Gas Turbines

In SUMTIME we are also working with sensor data from gas turbines. Compared to the domain of meteorology, the data densities from Gas Turbine are very high. The exact number depends upon the particular turbine being monitored, but as an order of magnitude we get hundreds of parameters measured continuously at one-second intervals. In this domain, we do not have a corpus of naturally-occurring human written summaries.

Figure 7. shows the plots of two parameters, TTXM (Exhaust temperature median corrected by average) and FQL1 (Liquid fuel flow magnetic pickup input). We asked two experts to describe the encircled portions in the plots. One expert described the encircled portion from the TTXM plot (the upper plot) as, "a step rise, followed by a domed rise and fall". We have come across many similar cases like this where we need to report shapes such as 'dome'. For these cases, we want

to use segmentation, but instead of using linear segments we intend to approximate data with curve segments.

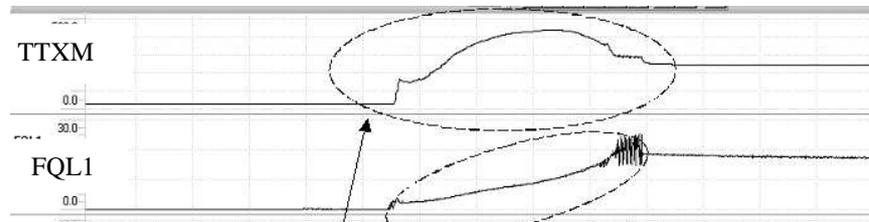


Figure 7. Plots of two example parameters (sensor data) from a Gas Turbine

The expert described the encircled portion of the FQL1 plot as “a ramped rise followed by oscillations before becoming steady”. In this case, we need to identify that the signal is oscillating, and we plan to explore if another variation of segmentation can be used. We feel that segmentation is potentially a more general process than linear piecewise segmentation. Particularly, we want to use segmentation at multiple levels; for example, perhaps first a coarse segmentation that would split the FQL1 signal into 3 portions, and then a more detailed analysis which would characterise the first portion as "ramped rise", the second portion as "oscillations", and the third portion as "steady". The analysis phase could use further segmentation but could also use other techniques, such as pattern-matching.

## 7. Conclusion And Future Work

We have presented our work on summarising time series data where an existing segmentation technique has been adapted to summarise weather data. This work demonstrates that knowledge acquired from domain experts and corpus studies can be used to modify an existing data analysis technique, in order to make it more effective for communication. We plan to continue our evaluation work further using other evaluation methods such as task-based evaluation and expert evaluation.

### Acknowledgements

Many thanks to our collaborators at WNI/Oceanroutes and Intelligent Applications, especially Ian Davy, Dave Selway, Rob Milne, and Jon Aylett; this work would not be possible without them! This project is supported by the UK Engineering and Physical Sciences Research Council (EPSRC), under grant GR/M76881.

### References

1. Sripada, S., Reiter, E., Hunter, J. and Yu, J. SUMTIME: Observation from KA for Weather Domain. Technical Report AUCS/TR0102. Dept. of Computing Science, University of Aberdeen, 2001.
2. Allen, J. and Perrault, C. R. Analyzing Intention in Utterances. *Artificial Intelligence* 1980; 26:1-33.

3. Sripada, S., Reiter, E., Hunter, J. and Yu, J. Modelling the task of Summarising Time Series Data using KA Techniques. In: Macintosh, A., Moulton, M. and Preece, A. (ed) Proceedings of ES-2001, 2001, pp 183 –196.
4. Scott, A., Clayton, J. and Gibson, E. A Practical Guide to Knowledge Acquisition. Addison-Wesley, 1991.
5. Reiter, E. and Dale, R. Building Natural Language Generation Systems. Cambridge University Press, 2000.
6. Boyd S, Detecting and Describing Patterns in Time-varying Data Using Wavelets. In: Lui, X. and Cohen, P. (ed) Advances in Intelligent Data Analysis: Reasoning About Data. Springer Verlag, 1997. (Lecture notes in computer science no. 1280)
7. Kukich, K. Design and implementation of a knowledge-based report generator. In: Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics (ACL-1983), 1983, pp 145-150.
8. Keogh, E. A fast and robust method for pattern matching in time-series data. In: Proceedings of WUSS-97, 1997.
9. Keogh, E., Chu, S., Hart, D. and Pazzani, M. An Online Algorithm for Segmenting Time Series. In: Proceedings of IEEE International Conference on Data Mining, 2001, pp 289-296.
10. Hunter, JRW and McIntosh, N. Knowledge-Based Event Detection in Complex Time Series Data. In: Horn, W et al. (ed) AIMDM'99: Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, Springer Verlag, 1999, pp271-280.
11. Goldberg, E., Driedger, N. and Kittredge, R. Using Natural-Language Processing to Produce Weather Reports. IEEE Expert 1994; 9:45-53.
12. Sripada, S., Reiter, E., Hunter J., and Yu, J. A Two-stage Model for Content Determination. In: Proceedings of ENLGW-2001, 2001, pp3-10.
13. Reiter, E., Robertson, R., Lennox, S., and Osman, L. Using a Randomised Controlled Clinical Trial to Evaluate an NLG System. In: Proceedings of ACL-2001, 2001, pp 434-441.